Multi-objective Optimization of a Parameterized VLIW Architecture

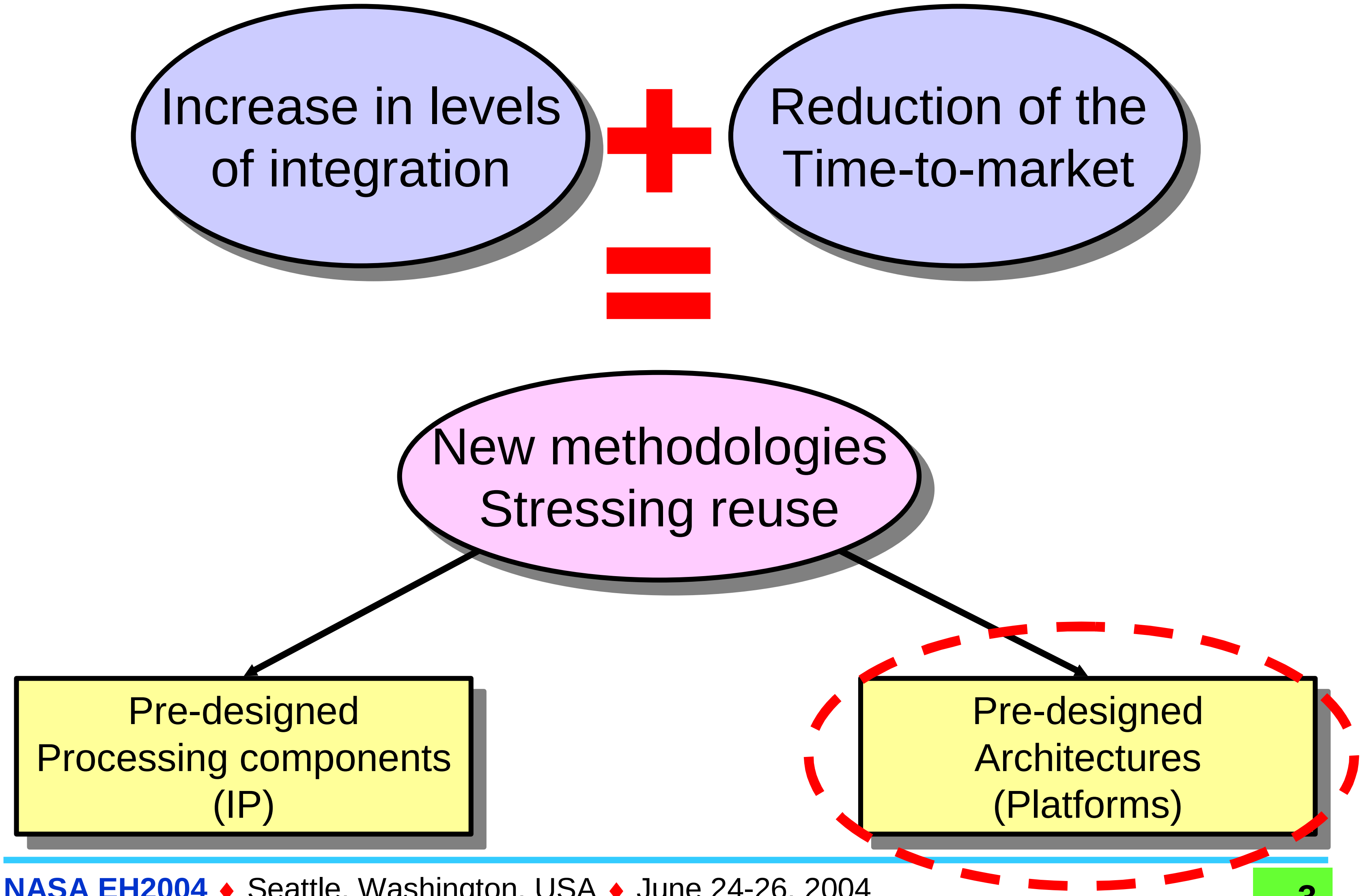
Giuseppe Ascia, Vincenzo Catania,
Maurizio Palesi and Davide Patti

DIIT - Università di Catania, Italy

Outline

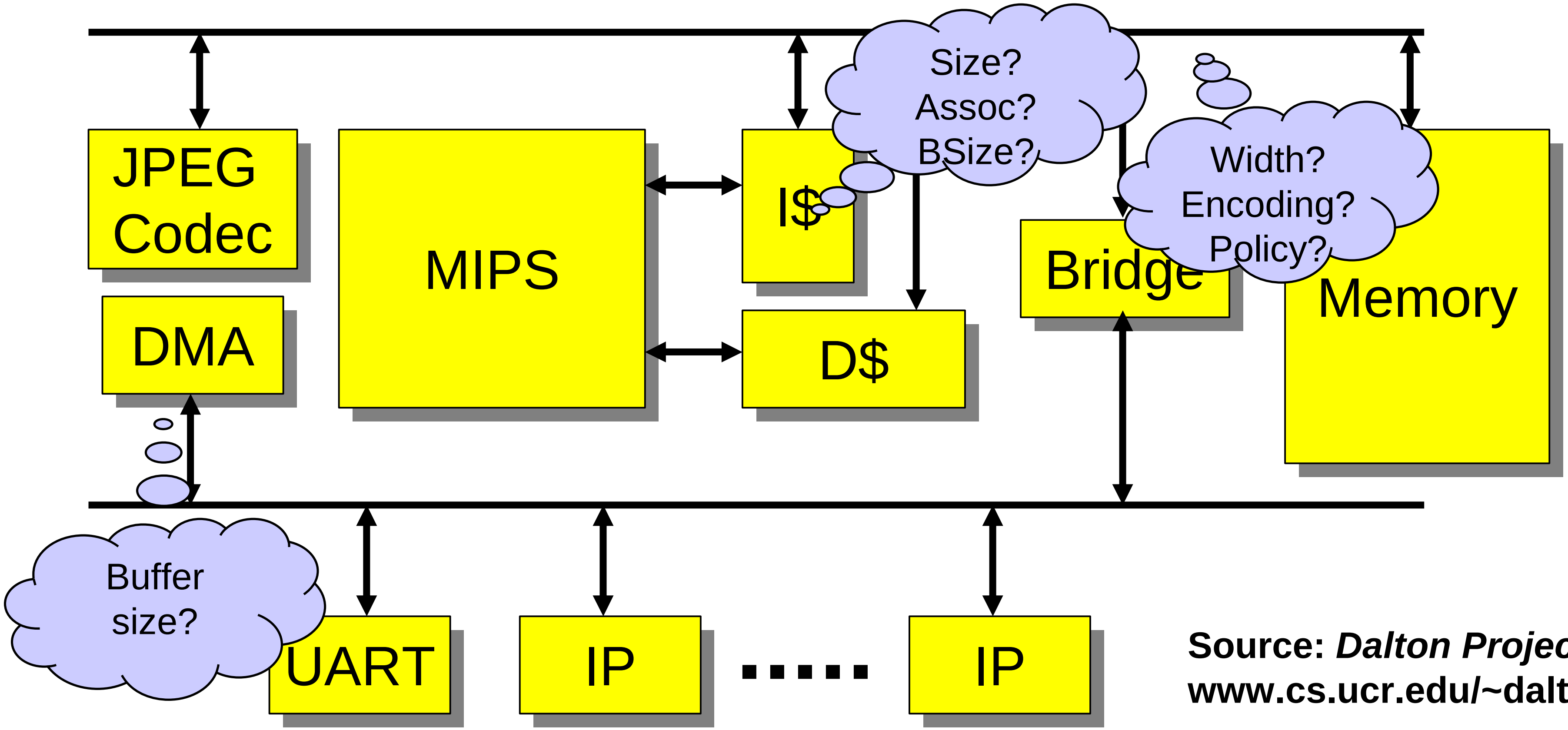
- Introduction
- Parameterized Platforms
- VLIW-based Parameterized Platform
- Estimation models
- Case Study : Exploration Methods
- Conclusions

Introduction



Goal of the Embedded System Designer

Optimally **configure** the platform **to meet** varied power, performance, cost, etc. requirements for a **fixed application**



Source: ***Dalton Project***
www.cs.ucr.edu/~dalton

Parameterized Platforms

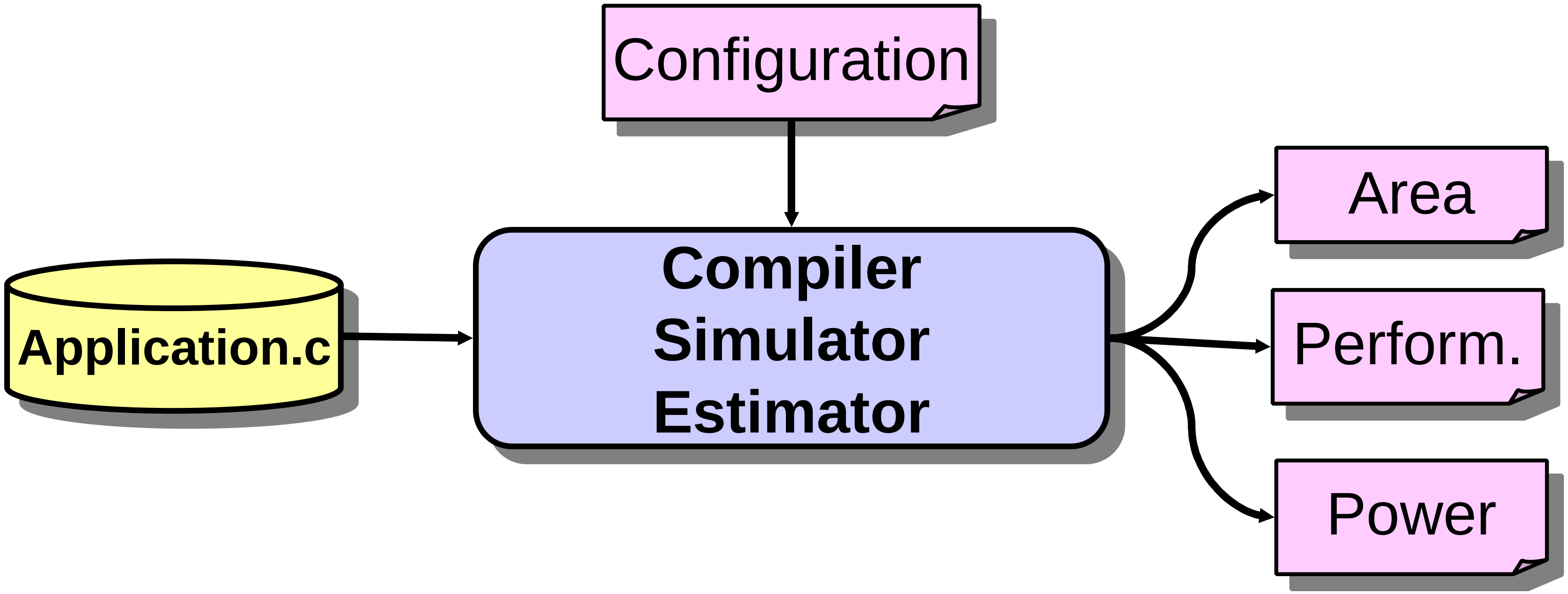
Terminology

- è A complete assigment of values to all the parameters is a **configuration**

- è A complete collection of all possible configurations is the Configuration Space (a.k.a. the Design Space)

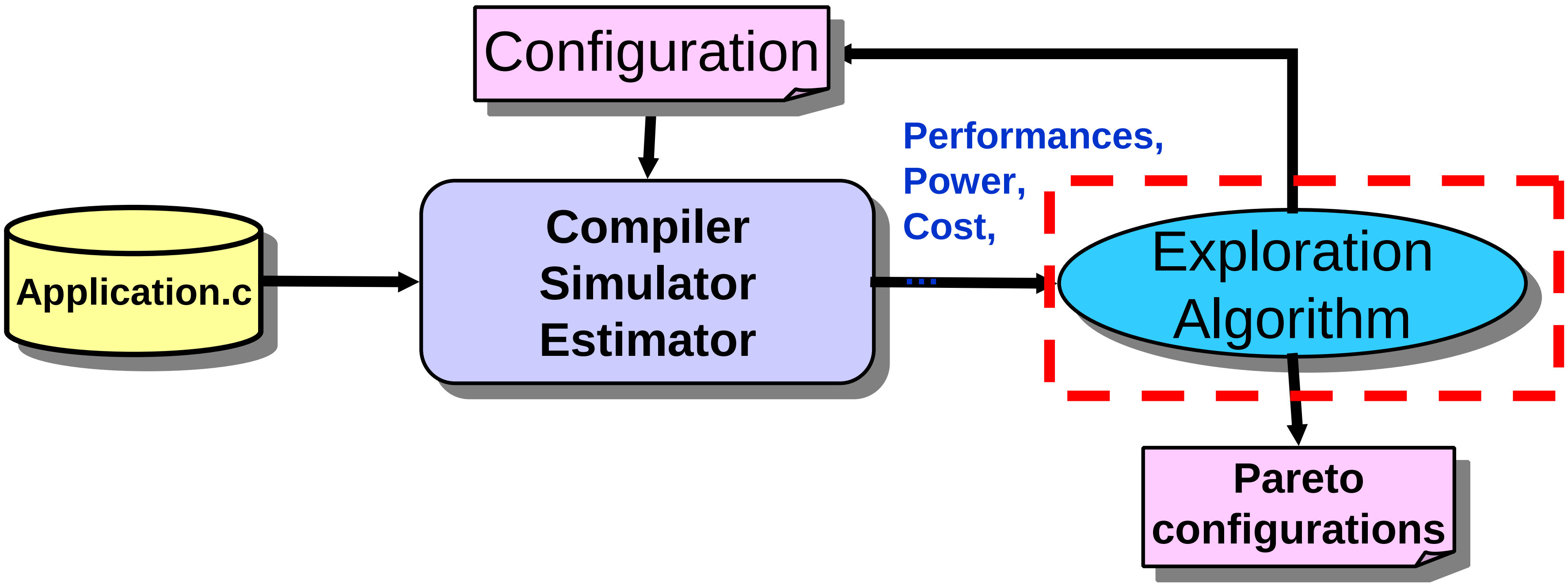
Required tools

Need for tools to quickly **evaluate the configurations** and thus system-level simulation and **estimation techniques**

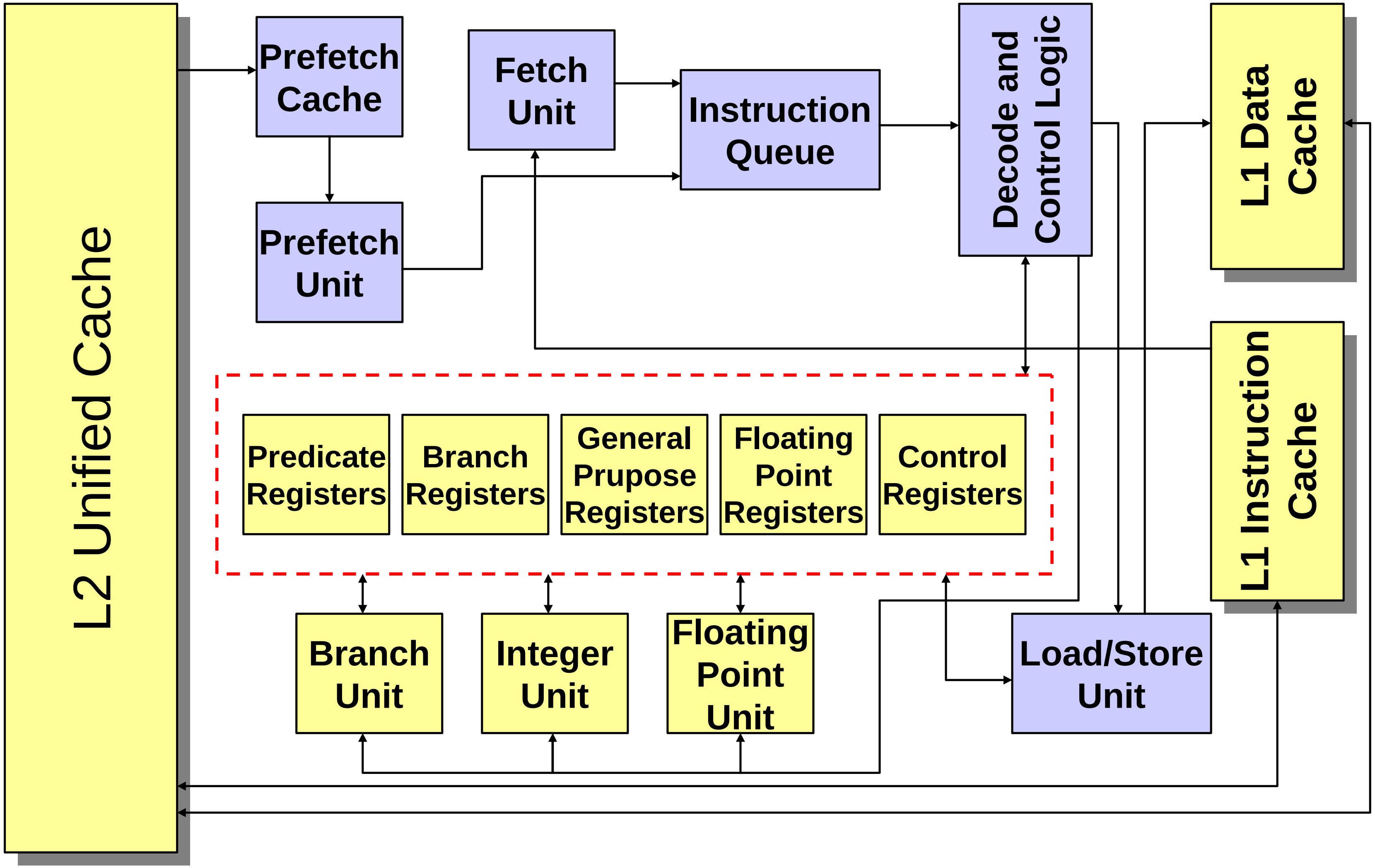


Required Tools

A Design Space Exploration strategy to find the configurations that represents the most promising trade-off between area/power/performance



Reference architecture (HPL-PD)



Configuration Space

Two parameter categories :

Processor :

- è Number of Registers in each register file (GPR,FPR,PR,CR,BTR)
- è Number of Functional Units of each type (IU,FPU,MU,BU)

Mem Hierarchy:

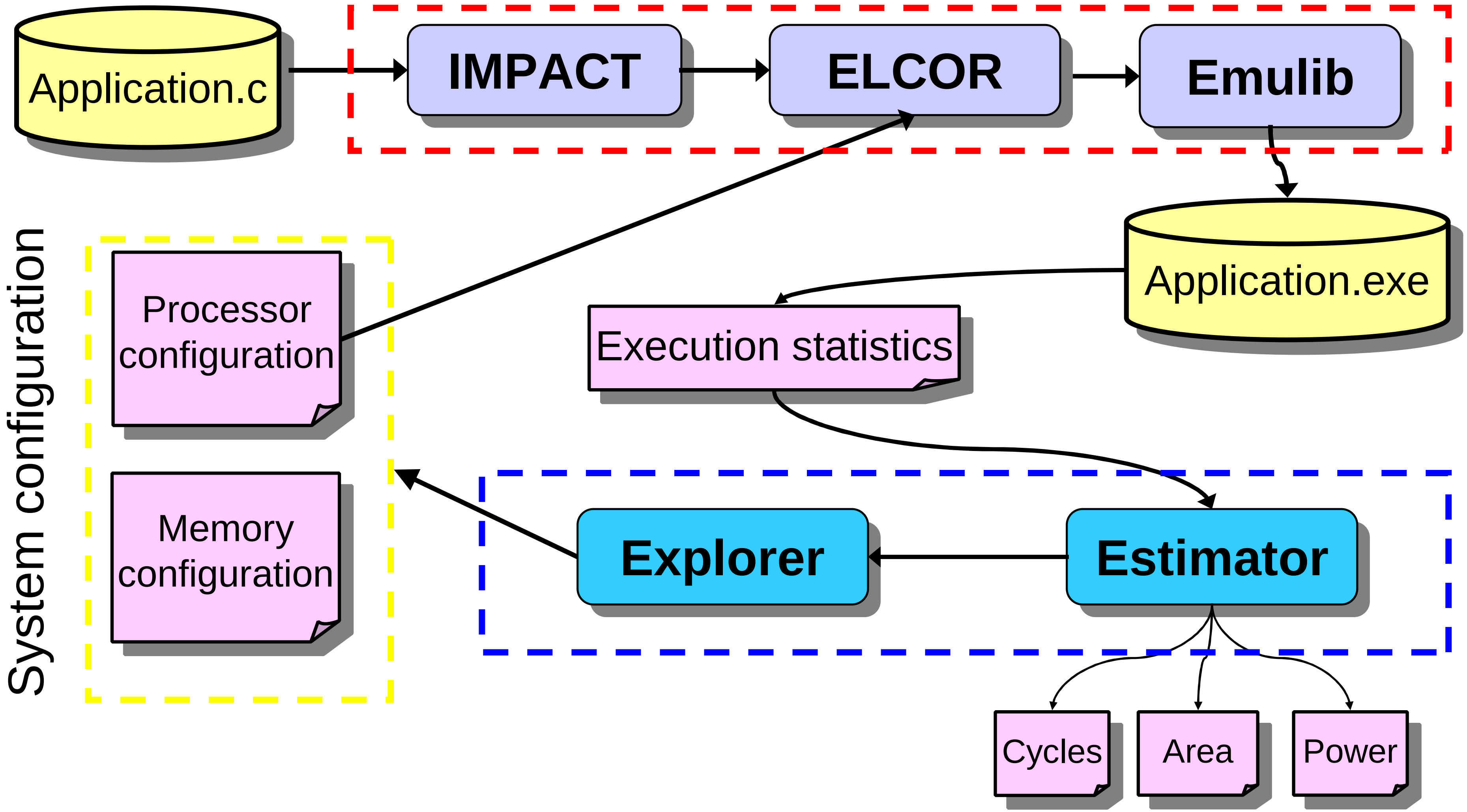
- è Size, Blocksize, Associativity for each of the caches (L1I,L1D,L2)

Total configuration space size: 1.47×10^{13}

Proposed platform: EPIC Explorer

- Interfacing to the Trimaran framework that provide VLIW compiler and simulator
- Realization of an estimator component that uses Trimaran output
- Realization of an exploration component that uses estimator output and implements multi-objective exploration of configuration space

The Data Flow



High level Estimation models

- It’s not possible to use very accurate but slow low level estimation on a such large configuration space
- Provide, early in the design cycle, a fast evaluation of the most promising configurations
- Discrete degree of accuracy (about 25%)
- Relative power savings beetween designs

Power estimation : processor

Cai-Lim model [*Cai and Lim '99*]

The architecture is subdivided into a set of FBU (Functional Block Unit)

For each FBU :

- è **Active power**: average power dissipated when the FBU is used
- è **Inactive power:** average power dissipated when the FBU is not used (due to static power consumption , usually from 10% to 50% of dynamic power)

From the exection statistic, we know how many cycles each FBU has been active and inactive

Power Estimation (caches)

CACTI model [*Jouppi et al.'99*]

The total amount of power dissipated by a cache is:

$P_{\$} = P_{bl} + P_{wl} + P_{out} + P_{ain}$

- P_{bl} preloading for eventual access, reading and writing
- P_{wl} selection by the driver of the wordlines for reading and writing operations
- P_{out} transitions of the external interconnection lines driven by the cache
- P_{ain} transitions in the address lines at the cache decoder input

Transitions are estimated using the dynamic statistics and the equations described by Kamble and Ghose [*ISLPED'97*]

Power Estimation (buses)

Bus lines transitions computed from the list of data/address memory accesses

$$P_{bus} = 0.5 \times (V_{dd})^2 \times \alpha \times f \times C_l$$

- V_{dd} supply voltage
- α switching activity
- f clock frequency
- C_l capacity of a bus line

Area Estimation

Processor

è Miyaoka *et al.* [ASPDAC'01]

ü $A_p = c_k + c_{RF} + c_{HU}$

- c_k kernel area: the nucleus of a processor that implemets the generic, essential functions (pipeline stages, buses for inst. & data memory, an ALU, a shifter etc. plus the increase in the area that the management of the additional components involves
- c_{RF} total contribution made by the register files
- c_{HU} contribution made by the hardware units (adders, comparators, shifters, …)

Cache

è **CACTI model [Jouppi et al.'99]**

Design Space Exploration

Even using fast high level estimation models, we need "intelligent" exploration strategies to avoid exhaustive evaluation of all possible configurations.

Two main goals of DSE:

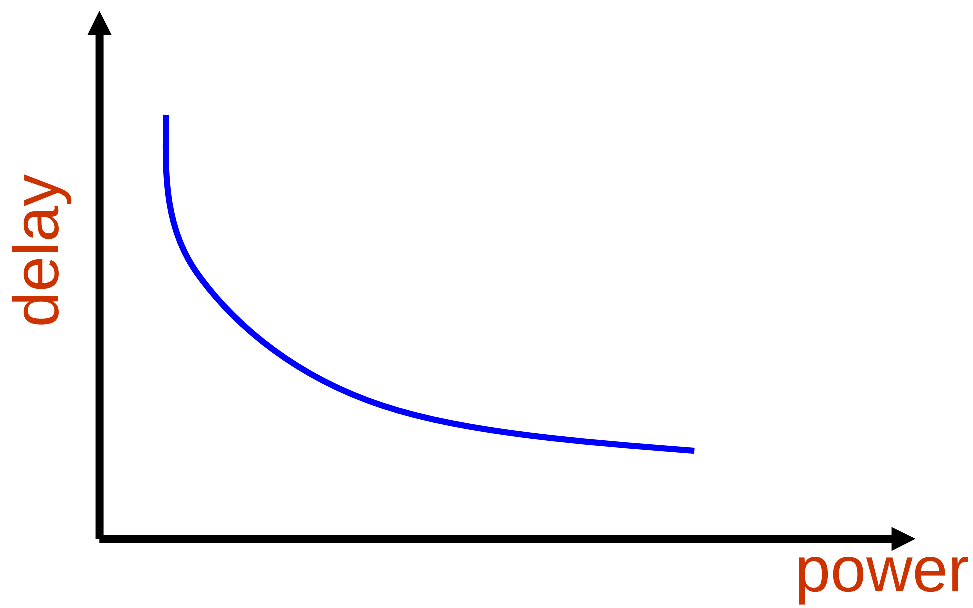
Accuracy: results similar to exhaustive exploration.

Efficiency: optimal pareto set searched in a reasonable time.

Multiobjective Exploration

Area, Power & Performance are objectives that often contrast each other

Pareto-optimal Set : there’s no single optimal solution, but a set of nondominated solutions.



Design Space Exploration

Implemented Algorithms :

Exhaustive: intuitive, simple and …unfeasible

Dependency analysis (dep), Givargis *et al.,* [TVLSI'02]

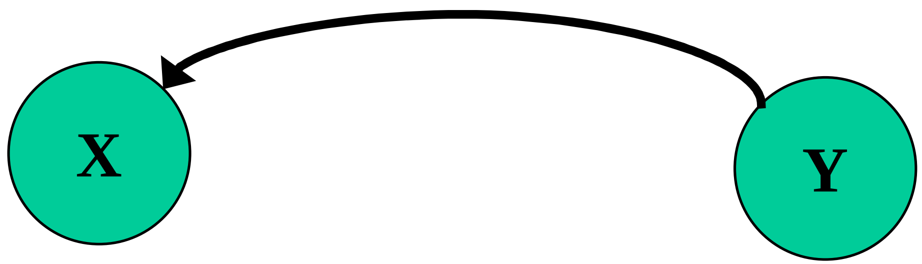
GA-based DSE (ga), Palesi *et al.*, [CODES'01]

Sensitivity Analysis, Fornaciari *et al.*, [DAES'02]

- è Pareto-based Sensitivity Analysis (pbsa), Palesi *et al.*, [VLSI-SOC'01]

Dependency analisys

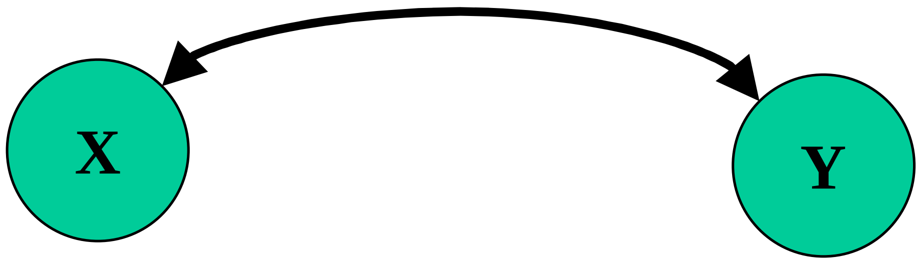
If the optimal value of a parameter X depends on the value of an other parameter Y, the X is said dependent from Y.



Optimal values of A must be computed once optimal values of B have been computed

Dependency analisys

If X depends on Y, and Y depends on X, parameters are defined interdependent.



The optimal values of interdependent parameters must be computed simultaneously.

DEP: How It Works

Interdependent parameters are grouped in clusters

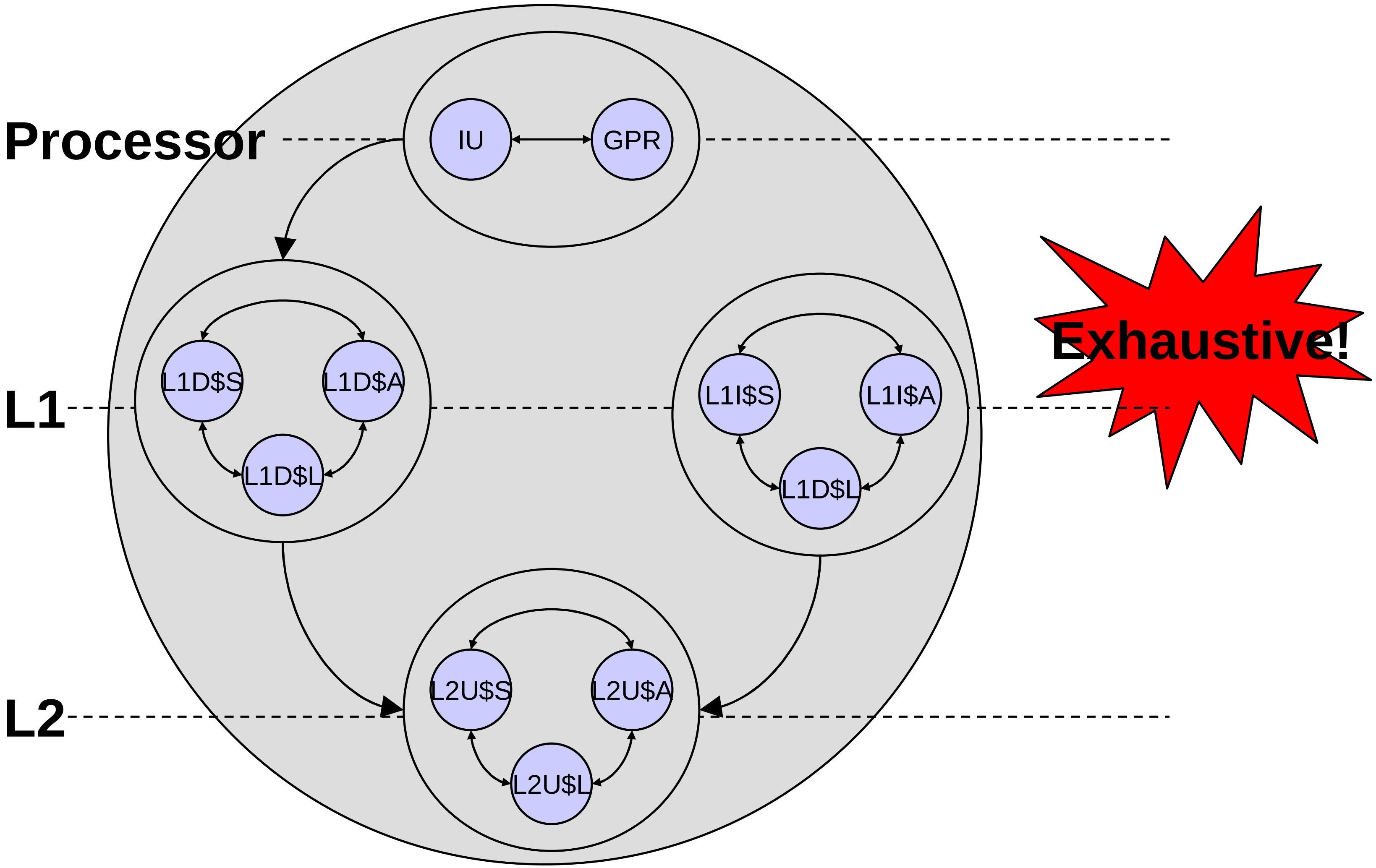
1st phase

- è Clusters are exhaustively explored with the aim to compute the local Pareto-optimal set (LPOS)

2nd phase

- è The LPOSs are merged and exhaustively searched to find the global Pareto-optimal set (GPOS)

DEP: Dependency graph



DEP

Advantages :

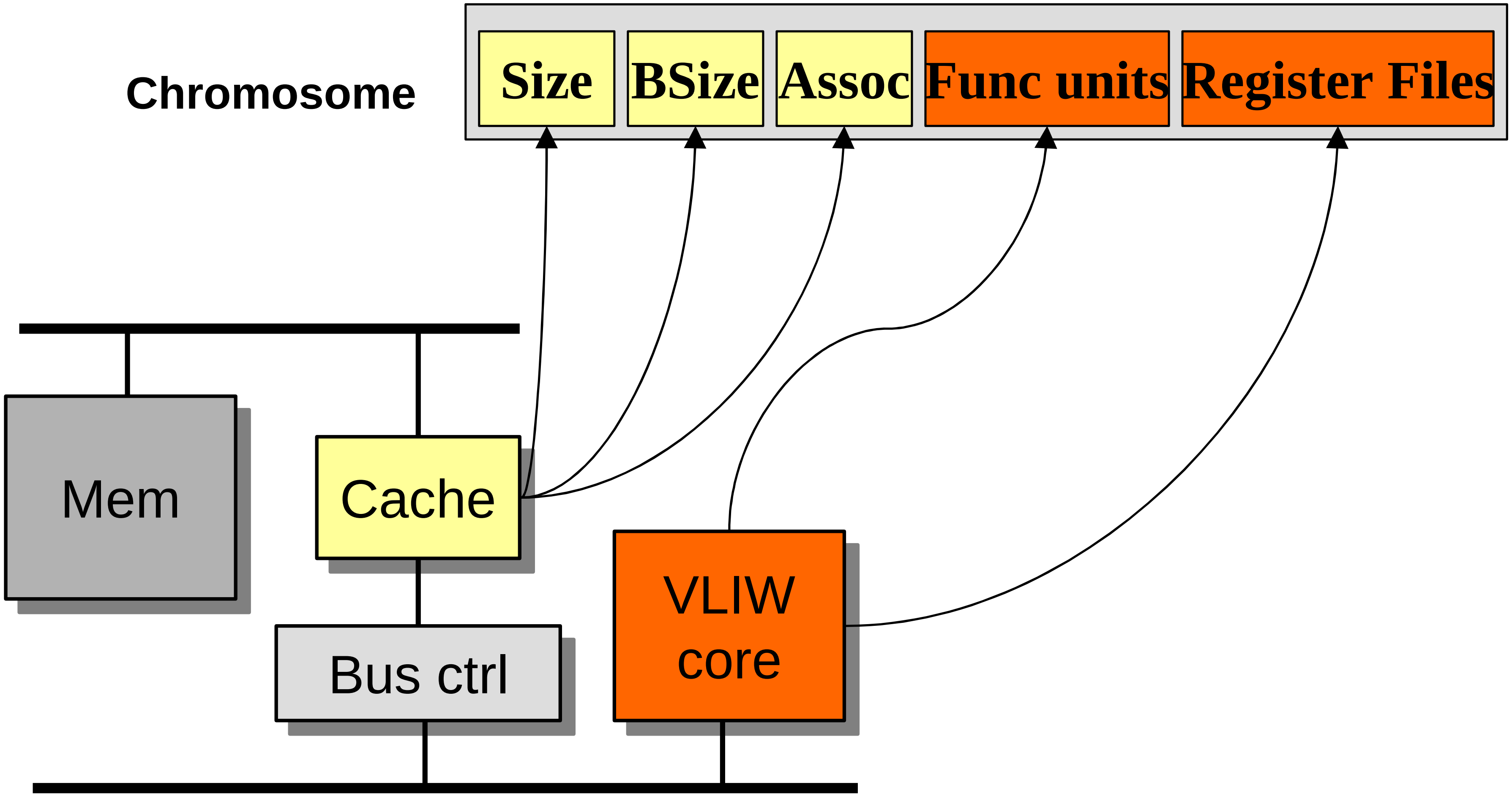
Virtual accuracy : If dependancy analisys is correct, its results are as accurate as exhaustive esploration

Disadvantages :

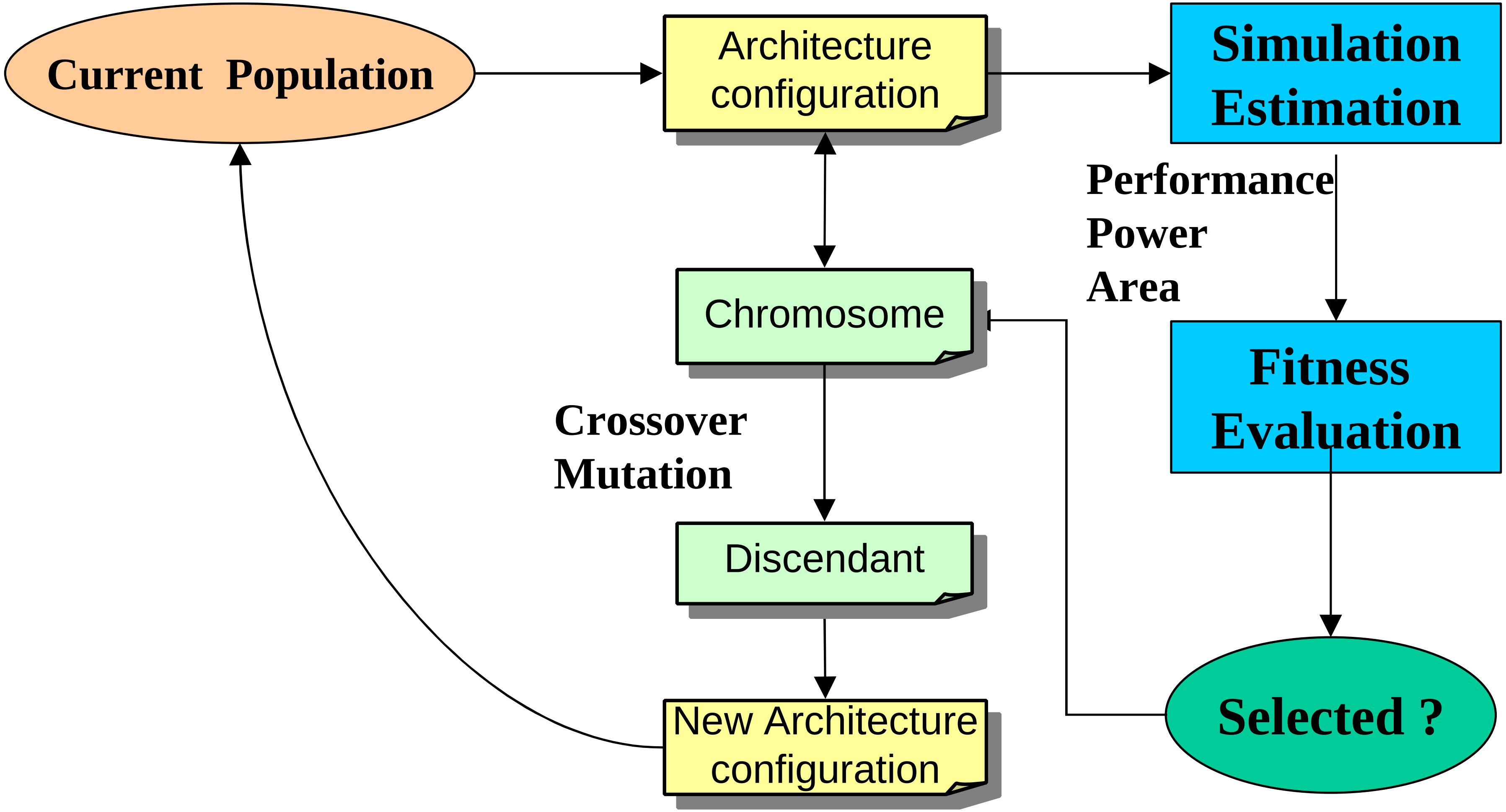
There no deterministic way to find all the parameter dependecies

Inefficiency : if interdependant parameters create big clusters, this approach is very time consuming

Genetic Approach



Genetic Approach Iteration



Multiobjective Fitness assignment

Strength Pareto Approach [Zitzler,Thiele]

From current population P , is extracted an ***external set P****, containing the nondominated configuration of P.

Fitness of P* element j : $f_j = n/(N+1)$

- N = total size of P
- n = # of P configurations dominated by j

Fitness of P element i: 1/S .

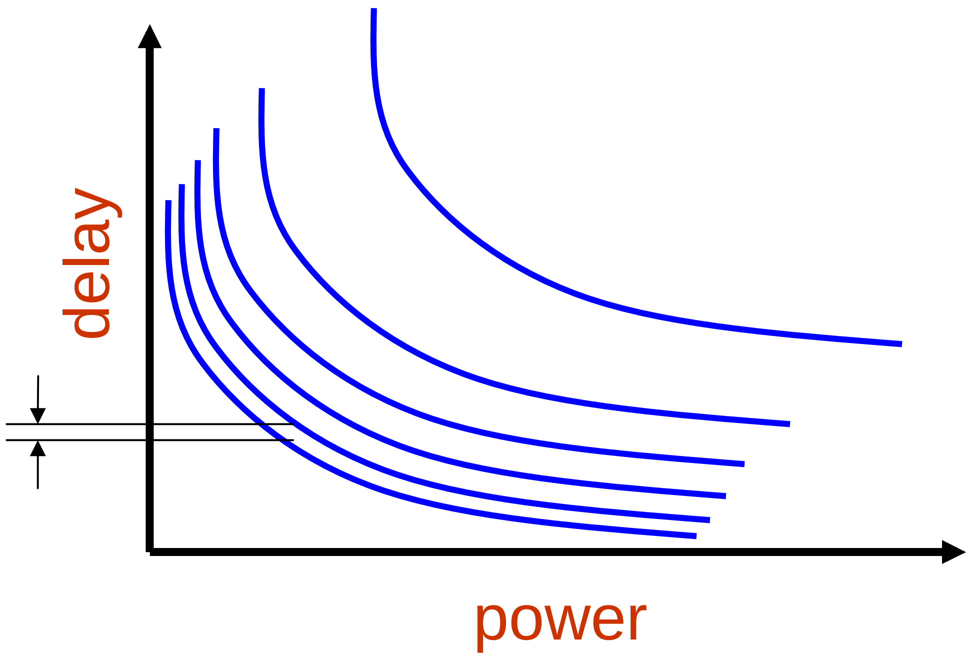
- S is the sum of the fitness values of the P* elements that dominates i

How Many Generations?

Fixed number of generations

Autostop criteria

- è Based on convergency



Multiobjective Genetic Approach

Dependency analisys is **not required**

Customizable (population size, crossover probability, mutation probability etc.)

Good efficiency : exploration time does not explode with larger parameters ranges

Good accuracy: in the subspaces where it was possible to compare it to ehxaustive exploration it showed very good accuracy even with 5-10 generations.

Design Space Exploration

Benchmark: JPEG (from the Motorola Powerstone benchmark suite)

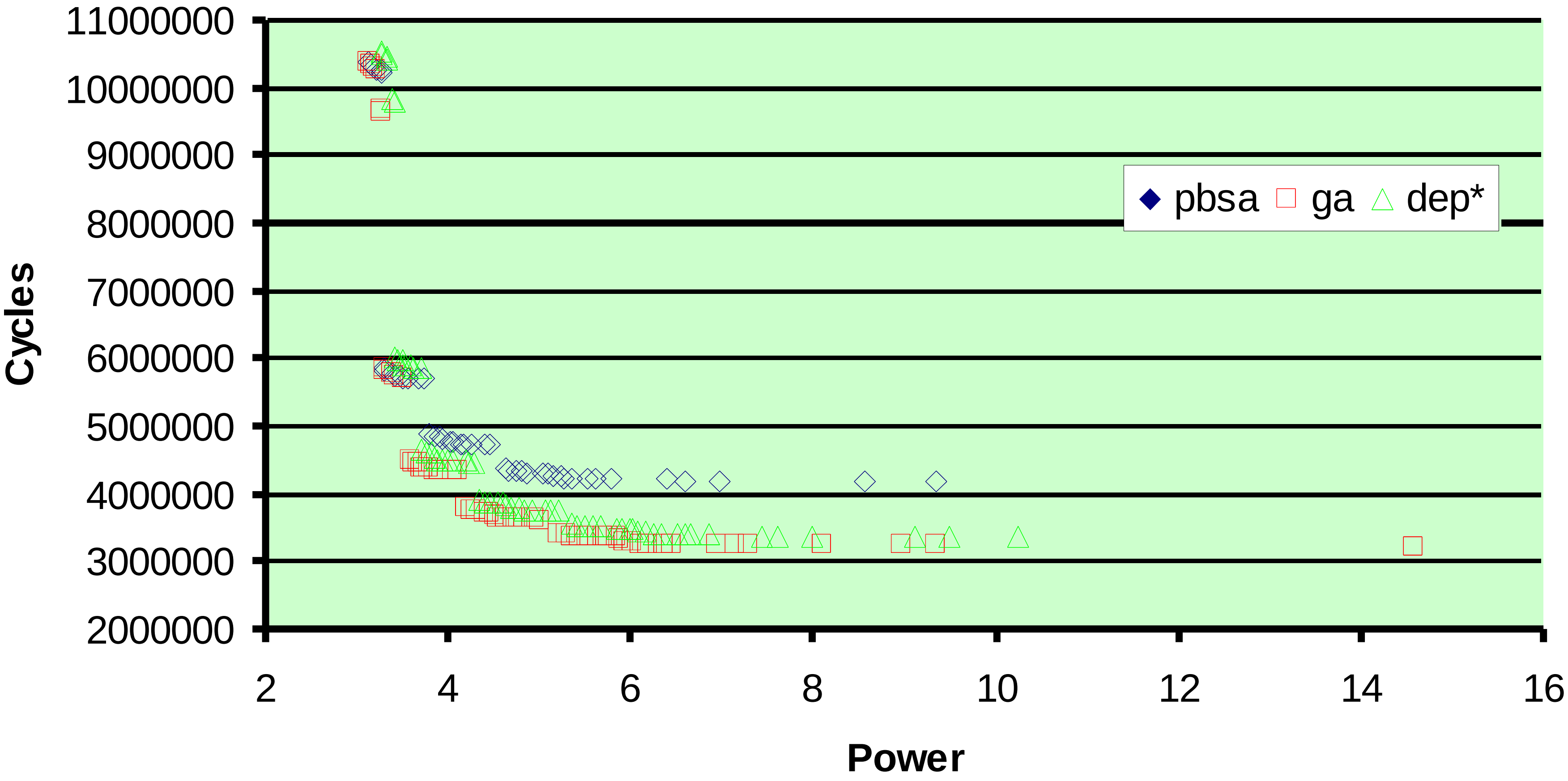
Parameters space

- L1 (8KB-32KB; 16B-64B; 1-2)
- L2 (64KB-256KB; 16B-64B; 2-8)
- Processor (IU: 1,2,3; GPR: 32,48,64)

Configuration space

- 78,732 configurations

Experimental Results



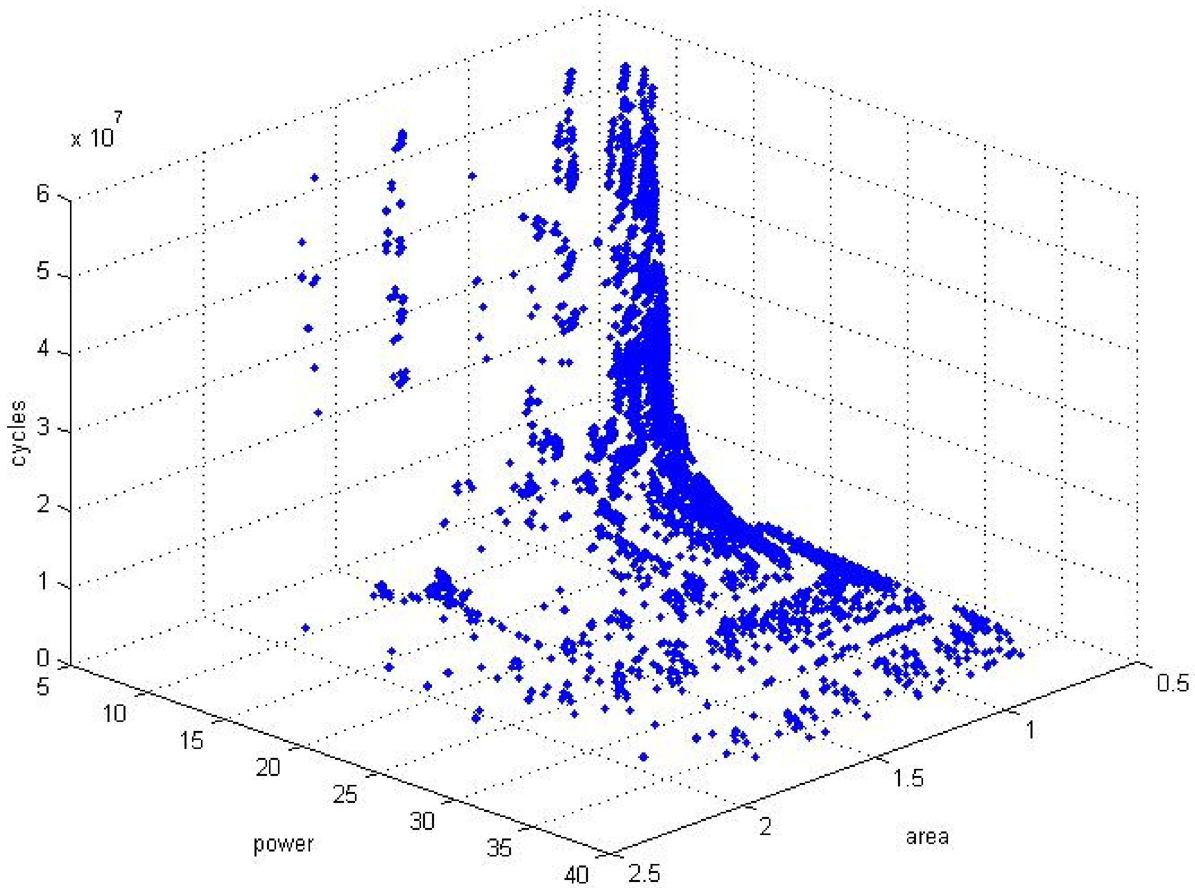
Experimental Results

On the total configuration space only genetic based approach exploration was possible.

Parameters :

- Initial population: 30 individuals
- Crossover probability: 0.8
- Mutation probability: 0.1
- Generations: 50

Experimental Results



Summarizing Table

Benchmark	Visited configurations	Elapsed Time	Pareto Set	Area trade-off	Power trade-off	Exec time Trade-off
Mpeg2decode (2D)	1137	47h	73	-	7x	6.8x
Jpeg (2D)	1012	17h	83	-	6x	8.2x
Mpeg2decode (3D)	1037	28h	175	3.8x	7x	9.6x

Conclusions

Parameterized VLIW-based Platform

- Power, Performance and Cost estimation
- Tuning the parameters for a given application
- Compare & Develop DSE strategies

Future developments

- New DSE methods
- Adding state-of-the-art estimation methods
- Open source : http://epic-explorer.sf.net